

MC9S12 Assembly Language Programming

Programming Model

Addressing Modes

Assembler Directives

HC12 Instructions

Flow Charts

1. Data Transfer and Manipulation Instructions — instructions which move and manipulate data (**S12CPUV2 Reference Manual**, Sections 5.3, 5.4, and 5.5).

- Load and Store — load copy of memory contents into a register; store copy of register contents into memory.

```
LDAA $2000 ; Copy contents of addr $2000 into A
STD 0,X    ; Copy contents of D to addrs X and X+1
```

- Transfer — copy contents of one register to another.

```
TBA                ; Copy B to A
TFR X,Y            ; Copy X to Y
```

- Exchange — exchange contents of two registers.

```
XGDX                ; Exchange contents of D and X
EXG A,B            ; Exchange contents of A and B
```

- Move — copy contents of one memory location to another.

```
MOVB $2000,$20A0    ; Copy byte at $2000 to $20A0
MOVW 2,X+,2,Y+      ; Copy two bytes from address held
                    ; in X to address held in Y
                    ; Add 2 to X and Y
```

2. Arithmetic Instructions — addition, subtraction, multiplication, division (**S12CPUV2 Reference Manual**, Sections 5.6, 5.8 and 5.12).

```
ABA                ; Add B to A; results in A
SUBD $20A1         ; Subtract contents of $20A1 from D
INX                ; Increment X by 1
MUL                ; Multiply A by B; results in D
```

3. Logic and Bit Instructions — perform logical operations (**S12CPUV2 Reference Manual**, Sections 5.9, 5.10, 5.11, 5.13 and 5.14).

- Logic Instructions

```
ANDA $2000 ; Logical AND of A with contents of $2000
EORB 2,X   ; Exclusive OR B with contents of address (X+2)
```

- Clear, Complement and Negate Instructions

```
NEG -2,X ; Negate (2's comp) contents of address (X-2)
CLRA     ; Clear Acc A
```

- Bit manipulate and test instructions — work with one bit of a register or memory.

```
BITA #$08          ; Check to see if Bit 3 of A is set
BSET $0002,$$18   ; Set bits 3 and 4 of address $002
```

- Shift and rotate instructions

```
LSLA              ; Logical shift left A
ASR $1000         ; Arithmetic shift right value at address $1000
```

4. Compare and test instructions — test contents of a register or memory (to see if zero, negative, etc.), or compare contents of a register to memory (to see if bigger than, etc.) (**S12CPUV2 Reference Manual**, Section 5.9).

```
TSTA             ; (A)-0 -- set flags accordingly
CPX  $$8000     ; (X) - $8000 -- set flags accordingly
```

5. Jump and Branch Instructions — Change flow of program (e.g., goto, it-then-else, switch-case) (**S12CPUV2 Reference Manual**, Sections 5.19, 5.20 and 5.21).

```
JMP L1          ; Start executing code at address label L1
BEQ L2          ; If Z bit set, go to label L2
DBNE X,L3       ; Decrement X; if X not 0 then goto L3
BRCLR $1A,$$80,L4 ; If bit 7 of addr $1A clear, go to label L4
JSR sub1        ; Jump to subroutine sub1
RTS             ; Return from subroutine
```

6. Interrupt Instructions — Initiate or terminate an interrupt call (**S12CPUV2 Reference Manual**, Section 5.22).

- Interrupt instructions

```
SWI             ; Initiate software interrupt
RTI             ; Return from interrupt
```

7. Index Manipulation Instructions — Put address into X, Y or SP, manipulate X, Y or SP (**S12CPUV2 Reference Manual**, Section 5.23).

ABX ; Add (B) to (X)
LEAX 5,Y ; Put address (Y) + 5 into X

8. Condition Code Instructions — change bits in Condition Code Register (**S12CPUV2 Reference Manual**, Section 5.26).

ANDCC #\$f0 ; Clear N, Z, C and V bits of CCR
SEV ; Set V bit of CCR

9. Stacking Instructions — push data onto and pull data off of stack (**S12CPUV2 Reference Manual**, Section 5.24).

PSHA ; Push contents of A onto stack
PULX ; Pull two top bytes of stack, put into X

10. Stop and Wait Instructions — put MC9S12 into low power mode (**S12CPUV2 Reference Manual**, Section 5.27).

STOP ; Put into lowest power mode
WAI ; Put into low power mode until next interrupt

11. Null Instructions

NOP ; No operation
BRN ; Branch never

12. Instructions we won't discuss or use — BCD arithmetic, fuzzy logic, minimum and maximum, multiply-accumulate, table interpolation (**S12CPUV2 Reference Manual**, Sections 5.7, 5.16, 5.17, and 5.18).

Disassembly of an MC9S12 Program

- It is sometimes useful to be able to convert MC9S12 op codes into mnemonics.
- For example, consider the hex code:

```

ADDR  DATA
-----
1000  C6 05 CE 20 00 E6 01 18 06 04 35 EE 3F

```

- To determine the instructions, use Tables A.2 through A.7 of the S12CPUV2 Reference Manual.
 - If the first byte of the instruction is anything other than \$18, use Sheet 1 of Table A.2 (Page 395). From this table, determine the number of bytes of the instruction and the addressing mode. For example, \$C6 is a two-byte instruction, the mnemonic is LDAB, and it uses the IMM addressing mode. Thus, the two bytes C6 05 is the op code for the instruction LDAB #\$05.
 - If the first byte is \$18, use Sheet 2 of Table A.2 (Page 396), and do the same thing. For example, 18 06 is a two byte instruction, the mnemonic is ABA, and it uses the INH addressing mode, so there is no operand. Thus, the two bytes 18 06 is the op code for the instruction ABA.
 - Indexed addressing mode is fairly complicated to disassemble. You need to use Table A.3 to determine the operand. For example, the op code \$E6 indicates LDAB indexed, and may use two to four bytes (one to three bytes in addition to the op code). The postbyte 01 indicates that the operand is 0,1, which is 5-bit constant offset, which takes only one additional byte. All 5-bit constant offset, pre and post increment and decrement, and register offset instructions use one additional byte. All 9-bit constant offset instructions use two additional bytes, with the second byte holding 8 bits of the 9 bit offset. (The 9th bit is a direction bit, which is held in the first postbyte.) All 16-bit constant offset instructions use three postbytes, with the 2nd and 3rd holding the 16-bit unsigned offset.
 - Transfer (TFR) and exchange (EXG) instructions all have the op code \$B7. Use Table A.5 to determine whether it is TFR or an EXG, and to determine which registers are being used. If the most significant bit of the postbyte is 0, the instruction is a transfer instruction.
 - Loop instructions (*Decrement and Branch*, *Increment and Branch*, and *Test and Branch*) all have the op code \$04. To determine which instruction the op code \$04 implies, and whether the branch is positive (forward) or negative (backward), use Table A.6. For example, in the sequence 04 35 EE, the 04 indicates a loop instruction. The 35 indicates it is a DBNE X instruction (decrement register X and branch if result is not equal to zero), and the direction is backward (negative). The EE indicates a branch of -18 bytes.
- Use up all the bytes for one instruction, then go on to the next instruction.

C6 05	=> LDAA #\$05	two-byte LDAA, IMM addressing mode
CE 20 00	=> LDX #\$2000	three-byte LDX, IMM addressing mode
E6 01	=> LDAB 1,X	two to four-byte LDAB, IDX addressing mode. Operand 01 => 1,X, a 5b constant offset which uses only one postbyte
18 06	=> ABA	two-byte ABA, INH addressing mode
04 35 EE	=> DBNE X,(-18)	three-byte loop instruction Postbyte 35 indicates DBNE X, negative
3F	=> SWI	one-byte SWI, INH addressing mode

Table A-2. CPU12 Opcode Map (Sheet 1 of 2)

00	5	10	1	20	3	30	3	40	3	50	1	60	3-6	70	4	80	3	90	3	A0	3-6	B0	3	C0	1	D0	3	E0	3-6	F0	3
BGND		ANDCC		BRA		PULX		NEGA		NEGB		NEG		NEG		SUBA		SUBA		SUBA		SUBA		SUBB		SUBB		SUBB		SUBB	
IH	1	IM	2	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
01	5	11	11	21	1	31	3	41	1	51	1	61	3-6	71	4	81	1	91	3	A1	3-6	B1	3	C1	1	D1	3	E1	3-6	F1	3
MEM		EDIV		BRN		PULY		COMA		COMB		COM		COM		CMPA		CMPA		CMPA		CMPA		CMPB		CMPB		CMPB		CMPB	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
02	1	12	11	22	3/1	32	3	42	1	52	1	62	3-6	72	4	82	1	92	3	A2	3-6	B2	3	C2	1	D2	3	E2	3-6	F2	3
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
03	1	13	3	23	3/1	33	3	43	1	53	1	63	3-6	73	4	83	2	93	3	A3	3-6	B3	3	C3	2	D3	3	E3	3-6	F3	3
DEY		EMUL		BLS		PULB		DECA		DECB		DEC		DEC		SUBD		SUBD		SUBD		SUBD		ADD		ADD		ADD		ADD	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
04	3	14	11	24	3/1	34	2	44	1	54	1	64	3-6	74	4	84	1	94	3	A4	3-6	B4	3	C4	1	D4	3	E4	3-6	F4	3
loop		ORCC		BCC		PSHX		LSRA		LSRB		LSR		LSR		ANDA		ANDA		ANDA		ANDA		ANDB		ANDB		ANDB		ANDB	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
05	3-6	15	4-7	25	3/1	35	2	45	1	55	1	65	3-6	75	4	85	1	95	3	A5	3-6	B5	3	C5	1	D5	3	E5	3-6	F5	3
JMP		JSR		BCS		PSHY		ROLA		ROLB		ROL		ROL		BITA		BITA		BITA		BITA		BITB		BITB		BITB		BITB	
ID	2-4	ID	2-4	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
06	3	16	4	26	3/1	36	2	46	1	56	1	66	3-6	76	4	86	1	96	3	A6	3-6	B6	3	C6	1	D6	3	E6	3-6	F6	3
JMP		JSR		BNE		PSHA		RORA		RORB		ROR		ROR		LDAA		LDAA		LDAA		LDAA		LDAB		LDAB		LDAB		LDAB	
EX	3	EX	3	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
07	4	17	4	27	3/1	37	2	47	1	57	1	67	3-6	77	4	87	1	97	3	A7	3-6	B7	3	C7	1	D7	3	E7	3-6	F7	3
BSR		JSR		BEG		PSHB		ASRA		ASRB		ASR		ASR		CLRA		TSTA		NOP		TFR/EXG		CLRB		TSTB		TST		TST	
RL	2	DI	2	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IH	1	IH	1	IH	1	IH	1	IH	1	IH	1	ID	2-4	EX	3
08	1	18	1	28	3/1	38	3	48	1	58	1	68	3-6	78	4	88	1	98	3	A8	3-6	B8	3	C8	1	D8	3	E8	3-6	F8	3
IH	1	Page 2		BVC		PULC		ASLA		ASLB		ASL		ASL		EORA		EORA		EORA		EORA		EORB		EORB		EORB		EORB	
IH	1			RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
09	1	19	2	29	3/1	39	2	49	1	59	1	69	12-4	79	3	89	1	99	3	A9	3-6	B9	3	C9	1	D9	3	E9	3-6	F9	3
DEX		LEAY		BVS		PSHC		LSRD		ASLD		CLR		CLR		ADCA		ADCA		ADCA		ADCA		ADCB		ADCB		ADCB		ADCB	
IH	1	ID	2-4	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
0A	17	1A	2	2A	3/1	3A	3	4A	17	5A	2	6A	12-4	7A	3	8A	1	9A	3	AA	3-6	BA	3	CA	1	DA	3	EA	3-6	FA	3
RTC		LEAX		BPL		PULD		CALL		STAA		STAA		STAA		ORAA		ORAA		ORAA		ORAA		ORAB		ORAB		ORAB		ORAB	
IH	1	ID	2-4	RL	2	IH	1	EX	4	DI	2	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
0B	18	1B	2	2B	3/1	3B	2	4B	17-10	5B	2	6B	12-4	7B	3	8B	1	9B	3	AB	3-6	BB	3	CB	1	DB	3	EB	3-6	FB	3
RTI		LEAS		BMI		PSHD		CALL		STAB		STAB		STAB		ADDA		ADDA		ADDA		ADDA		ADDB		ADDB		ADDB		ADDB	
IH	1	ID	2-4	RL	2	IH	1	ID	2-5	DI	2	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
0C	4-6	1C	4	2C	3/1	3C	1+5	4C	4	5C	2	6C	12-4	7C	3	8C	2	9C	3	AC	3-6	BC	3	CC	2	DC	3	EC	3-6	FC	3
BSET		BSET		BGE		wavr		BSET		STD		STD		STD		CPD		CPD		CPD		CPD		LDD		LDD		LDD		LDD	
ID	3-5	EX	4	RL	2	SP	1	DI	3	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
0D	4-6	1D	4	2D	3/1	3D	5	4D	4	5D	2	6D	12-4	7D	3	8D	2	9D	3	AD	3-6	BD	3	CD	2	DD	3	ED	3-6	FD	3
BCLR		BCLR		BLT		RTS		BCLR		STY		STY		STY		CPY		CPY		CPY		CPY		LDY		LDY		LDY		LDY	
ID	3-5	EX	4	RL	2	IH	1	DI	3	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
0E	14-6	1E	5	2E	3/1	3E	1+7	4E	4	5E	2	6E	12-4	7E	3	8E	2	9E	3	AE	3-6	BE	3	CE	2	DE	3	EE	3-6	FE	3
BRSET		BRSET		BGT		WAI		BRSET		STX		STX		STX		CPX		CPX		CPX		CPX		LDX		LDX		LDX		LDX	
ID	4-6	EX	5	RL	2	IH	1	DI	4	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
0F	14-6	1F	5	2F	3/1	3F	9	4F	4	5F	2	6F	12-4	7F	3	8F	2	9F	3	AF	3-6	BF	3	CF	2	DF	3	EF	3-6	FF	3
BRCLR		BRCLR		BLE		SWI		BRCLR		STS		STS		STS		CPS		CPS		CPS		CPS		LDS		LDS		LDS		LDS	
ID	4-6	EX	5	RL	2	IH	1	DI	4	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2

Key to Table A-2

- Opcode → 00 5 ← Number of HCS12 cycles (‡ indicates HC12 different)
- Mnemonic → BGND ←
- Address Mode → IH 1 ← Number of bytes

Table A-2. CPU12 Opcode Map (Sheet 2 of 2)

00	4	10	12	20	4	30	10	40	10	50	10	60	10	70	10	80	10	90	10	A0	10	B0	10	C0	10	D0	10	E0	10	F0	10
MOVW		IDIV		LBRA		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-ID	5	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
01	5	11	12	21	3	31	10	41	10	51	10	61	10	71	10	81	10	91	10	A1	10	B1	10	C1	10	D1	10	E1	10	F1	10
MOVW		FDIV		LBRN		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-ID	5	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
02	5	12	13	22	4/3	32	10	42	10	52	10	62	10	72	10	82	10	92	10	A2	10	B2	10	C2	10	D2	10	E2	10	F2	10
MOVW		EMACS		LBHI		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-ID	4	SP	4	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
03	5	13	3	23	4/3	33	10	43	10	53	10	63	10	73	10	83	10	93	10	A3	10	B3	10	C3	10	D3	10	E3	10	F3	10
MOVW		EMULS		LBSL		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-EX	6	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
04	6	14	12	24	4/3	34	10	44	10	54	10	64	10	74	10	84	10	94	10	A4	10	B4	10	C4	10	D4	10	E4	10	F4	10
MOVW		EDIVS		LBCC		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-EX	6	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
05	5	15	12	25	4/3	35	10	45	10	55	10	65	10	75	10	85	10	95	10	A5	10	B5	10	C5	10	D5	10	E5	10	F5	10
MOVW		IDIVS		LBCS		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-EX	5	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
06	2	16	2	26	4/3	36	10	46	10	56	10	66	10	76	10	86	10	96	10	A6	10	B6	10	C6	10	D6	10	E6	10	F6	10
ABA		SBA		LBNE		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
07	3	17	2	27	4/3	37	10	47	10	57	10	67	10	77	10	87	10	97	10	A7	10	B7	10	C7	10	D7	10	E7	10	F7	10
DAA		CBA		LBEO		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
08	4	18	4-7	28	4/3	38	10	48	10	58	10	68	10	78	10	88	10	98	10	A8	10	B8	10	C8	10	D8	10	E8	10	F8	10
MOVW		MAXA		LBVC		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-ID	4	ID	3-5	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
09	5	19	4-7	29	4/3	39	10	49	10	59	10	69	10	79	10	89	10	99	10	A9	10	B9	10	C9	10	D9	10	E9	10	F9	10
MOVW		MINA		LBVS		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-ID	5	ID	3-5	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0A	5	1A	4-7	2A	4/3	3A	†3n	4A	10	5A	10	6A	10	7A	10	8A	10	9A	10	AA	10	BA	10	CA	10	DA	10	EA	10	FA	10
MOVW		EMAXD		LBPL		REV		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-ID	4	ID	3-5	RL	4	SP	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0B	4	1B	4-7	2B	4/3	3B	†5n/3n	4B	10	5B	10	6B	10	7B	10	8B	10	9B	10	AB	10	BB	10	CB	10	DB	10	EB	10	FB	10
MOVW		EMIND		LBMI		REVV		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-EX	5	ID	3-5	RL	4	SP	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0C	6	1C	4-7	2C	4/3	3C	†7B	4C	10	5C	10	6C	10	7C	10	8C	10	9C	10	AC	10	BC	10	CC	10	DC	10	EC	10	FC	10
MOVW		MAXM		LBGE		WAV		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-EX	6	ID	3-5	RL	4	SP	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0D	5	1D	D4-7	2D	4/3	3D	†6	4D	10	5D	10	6D	10	7D	10	8D	10	9D	10	AD	10	BD	10	CD	10	DD	10	ED	10	FD	10
MOVW		MINM		LBTL		TBL		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-EX	5	ID	3-5	RL	4	ID	3	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0E	2	1E	4-7	2E	4/3	3E	†8	4E	10	5E	10	6E	10	7E	10	8E	10	9E	10	AE	10	BE	10	CE	10	DE	10	EE	10	FE	10
TAB		EMAXM		LBGT		STOP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	ID	3-5	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0F	2	1F	4-7	2F	4/3	3F	10	4F	10	5F	10	6F	10	7F	10	8F	10	9F	10	AF	10	BF	10	CF	10	DF	10	EF	10	FF	10
TBA		EMINM		LBLE		ETBL		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	ID	3-5	RL	4	ID	3	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2

* The opcode \$04 (on sheet 1 of 2) corresponds to one of the loop primitive instructions DBEO, DBNE, IBEO, IBNE, TBEQ, or TBNE.

† Refer to instruction summary for more information.

‡ Refer to instruction summary for different HC12 cycle count.

Page 2: When the CPU encounters a page 2 opcode (\$18 on page 1 of the opcode map), it treats the next byte of object code as a page 2 instruction opcode.

Table A-3. Indexed Addressing Mode Postbyte Encoding (xb)

00	0,X 5b const	10	-16,X 5b const	20	1,+X pre-inc	30	1,X+ post-inc	40	0,Y 5b const	50	-16,Y 5b const	60	1,+Y pre-inc	70	1,Y+ post-inc	80	0,SP 5b const	90	-16,SP 5b const	A0	1,+SP pre-inc	B0	1,SP+ post-inc	C0	0,PC 5b const	D0	-16,PC 5b const	E0	n,X 9b const	F0	n,SP 9b const
01	1,X 5b const	11	-15,X 5b const	21	2,+X pre-inc	31	2,X+ post-inc	41	1,Y 5b const	51	-15,Y 5b const	61	2,+Y pre-inc	71	2,Y+ post-inc	81	1,SP 5b const	91	-15,SP 5b const	A1	2,+SP pre-inc	B1	2,SP+ post-inc	C1	1,PC 5b const	D1	-15,PC 5b const	E1	-n,X 9b const	F1	-n,SP 9b const
02	2,X 5b const	12	-14,X 5b const	22	3,+X pre-inc	32	3,X+ post-inc	42	2,Y 5b const	52	-14,Y 5b const	62	3,+Y pre-inc	72	3,Y+ post-inc	82	2,SP 5b const	92	-14,SP 5b const	A2	3,+SP pre-inc	B2	3,SP+ post-inc	C2	2,PC 5b const	D2	-14,PC 5b const	E2	n,X 16b const	F2	n,SP 16b const
03	3,X 5b const	13	-13,X 5b const	23	4,+X pre-inc	33	4,X+ post-inc	43	3,Y 5b const	53	-13,Y 5b const	63	4,+Y pre-inc	73	4,Y+ post-inc	83	3,SP 5b const	93	-13,SP 5b const	A3	4,+SP pre-inc	B3	4,SP+ post-inc	C3	3,PC 5b const	D3	-13,PC 5b const	E3	[n,X] 16b indir	F3	[n,SP] 16b indir
04	4,X 5b const	14	-12,X 5b const	24	5,+X pre-inc	34	5,X+ post-inc	44	4,Y 5b const	54	-12,Y 5b const	64	5,+Y pre-inc	74	5,Y+ post-inc	84	4,SP 5b const	94	-12,SP 5b const	A4	5,+SP pre-inc	B4	5,SP+ post-inc	C4	4,PC 5b const	D4	-12,PC 5b const	E4	A,X A offset	F4	A,SP A offset
05	5,X 5b const	15	-11,X 5b const	25	6,+X pre-inc	35	6,X+ post-inc	45	5,Y 5b const	55	-11,Y 5b const	65	6,+Y pre-inc	75	6,Y+ post-inc	85	5,SP 5b const	95	-11,SP 5b const	A5	6,+SP pre-inc	B5	6,SP+ post-inc	C5	5,PC 5b const	D5	-11,PC 5b const	E5	B,X B offset	F5	B,SP B offset
06	6,X 5b const	16	-10,X 5b const	26	7,+X pre-inc	36	7,X+ post-inc	46	6,Y 5b const	56	-10,Y 5b const	66	7,+Y pre-inc	76	7,Y+ post-inc	86	6,SP 5b const	96	-10,SP 5b const	A6	7,+SP pre-inc	B6	7,SP+ post-inc	C6	6,PC 5b const	D6	-10,PC 5b const	E6	D,X D offset	F6	D,SP D offset
07	7,X 5b const	17	-9,X 5b const	27	8,+X pre-inc	37	8,X+ post-inc	47	7,Y 5b const	57	-9,Y 5b const	67	8,+Y pre-inc	77	8,Y+ post-inc	87	7,SP 5b const	97	-9,SP 5b const	A7	8,+SP pre-inc	B7	8,SP+ post-inc	C7	7,PC 5b const	D7	-9,PC 5b const	E7	[D,X] D indirect	F7	[D,SP] D indirect
08	8,X 5b const	18	-8,X 5b const	28	8,-X pre-dec	38	8,X- post-dec	48	8,Y 5b const	58	-8,Y 5b const	68	8,-Y pre-dec	78	8,Y- post-dec	88	8,SP 5b const	98	-8,SP 5b const	A8	8,-SP pre-dec	B8	8,SP- post-dec	C8	8,PC 5b const	D8	-8,PC 5b const	E8	n,Y 9b const	F8	n,PC 9b const
09	9,X 5b const	19	-7,X 5b const	29	7,-X pre-dec	39	7,X- post-dec	49	9,Y 5b const	59	-7,Y 5b const	69	7,-Y pre-dec	79	7,Y- post-dec	89	9,SP 5b const	99	-7,SP 5b const	A9	7,-SP pre-dec	B9	7,SP- post-dec	C9	9,PC 5b const	D9	-7,PC 5b const	E9	-n,Y 9b const	F9	-n,PC 9b const
0A	10,X 5b const	1A	-6,X 5b const	2A	6,-X pre-dec	3A	6,X- post-dec	4A	10,Y 5b const	5A	-6,Y 5b const	6A	6,-Y pre-dec	7A	6,Y- post-dec	8A	10,SP 5b const	9A	-6,SP 5b const	AA	6,-SP pre-dec	BA	6,SP- post-dec	CA	10,PC 5b const	DA	-6,PC 5b const	EA	n,Y 16b const	FA	n,PC 16b const
0B	11,X 5b const	1B	-5,X 5b const	2B	5,-X pre-dec	3B	5,X- post-dec	4B	11,Y 5b const	5B	-5,Y 5b const	6B	5,-Y pre-dec	7B	5,Y- post-dec	8B	11,SP 5b const	9B	-5,SP 5b const	AB	5,-SP pre-dec	BB	5,SP- post-dec	CB	11,PC 5b const	DB	-5,PC 5b const	EB	[n,Y] 16b indir	FB	[n,PC] 16b indir
0C	12,X 5b const	1C	-4,X 5b const	2C	4,-X pre-dec	3C	4,X- post-dec	4C	12,Y 5b const	5C	-4,Y 5b const	6C	4,-Y pre-dec	7C	4,Y- post-dec	8C	12,SP 5b const	9C	-4,SP 5b const	AC	4,-SP pre-dec	BC	4,SP- post-dec	CC	12,PC 5b const	DC	-4,PC 5b const	EC	A,Y A offset	FC	A,PC A offset
0D	13,X 5b const	1D	-3,X 5b const	2D	3,-X pre-dec	3D	3,X- post-dec	4D	13,Y 5b const	5D	-3,Y 5b const	6D	3,-Y pre-dec	7D	3,Y- post-dec	8D	13,SP 5b const	9D	-3,SP 5b const	AD	3,-SP pre-dec	BD	3,SP- post-dec	CD	13,PC 5b const	DD	-3,PC 5b const	ED	B,Y B offset	FD	B,PC B offset
0E	14,X 5b const	1E	-2,X 5b const	2E	2,-X pre-dec	3E	2,X- post-dec	4E	14,Y 5b const	5E	-2,Y 5b const	6E	2,-Y pre-dec	7E	2,Y- post-dec	8E	14,SP 5b const	9E	-2,SP 5b const	AE	2,-SP pre-dec	BE	2,SP- post-dec	CE	14,PC 5b const	DE	-2,PC 5b const	EE	D,Y D offset	FE	D,PC D offset
0F	15,X 5b const	1F	-1,X 5b const	2F	1,-X pre-dec	3F	1,X- post-dec	4F	15,Y 5b const	5F	-1,Y 5b const	6F	1,-Y pre-dec	7F	1,Y- post-dec	8F	15,SP 5b const	9F	-1,SP 5b const	AF	1,-SP pre-dec	BF	1,SP- post-dec	CF	15,PC 5b const	DF	-1,PC 5b const	EF	[D,Y] D indirect	FF	[D,PC] D indirect

Key to Table A-3

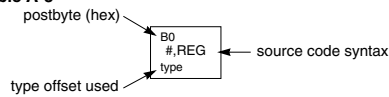


Table A-5. Transfer and Exchange Postbyte Encoding

TRANSFERS									
↓ LS	MS⇒	0	1	2	3	4	5	6	7
0		A ⇒ A	B ⇒ A	CCR ⇒ A	TMP3 _L ⇒ A	B ⇒ A	X _L ⇒ A	Y _L ⇒ A	SP _L ⇒ A
1		A ⇒ B	B ⇒ B	CCR ⇒ B	TMP3 _L ⇒ B	B ⇒ B	X _L ⇒ B	Y _L ⇒ B	SP _L ⇒ B
2		A ⇒ CCR	B ⇒ CCR	CCR ⇒ CCR	TMP3 _L ⇒ CCR	B ⇒ CCR	X _L ⇒ CCR	Y _L ⇒ CCR	SP _L ⇒ CCR
3		sex:A ⇒ TMP2	sex:B ⇒ TMP2	sex:CCR ⇒ TMP2	TMP3 ⇒ TMP2	D ⇒ TMP2	X ⇒ TMP2	Y ⇒ TMP2	SP ⇒ TMP2
4		sex:A ⇒ D SEX A,D	sex:B ⇒ D SEX B,D	sex:CCR ⇒ D SEX CCR,D	TMP3 ⇒ D	D ⇒ D	X ⇒ D	Y ⇒ D	SP ⇒ D
5		sex:A ⇒ X SEX A,X	sex:B ⇒ X SEX B,X	sex:CCR ⇒ X SEX CCR,X	TMP3 ⇒ X	D ⇒ X	X ⇒ X	Y ⇒ X	SP ⇒ X
6		sex:A ⇒ Y SEX A,Y	sex:B ⇒ Y SEX B,Y	sex:CCR ⇒ Y SEX CCR,Y	TMP3 ⇒ Y	D ⇒ Y	X ⇒ Y	Y ⇒ Y	SP ⇒ Y
7		sex:A ⇒ SP SEX A,SP	sex:B ⇒ SP SEX B,SP	sex:CCR ⇒ SP SEX CCR,SP	TMP3 ⇒ SP	D ⇒ SP	X ⇒ SP	Y ⇒ SP	SP ⇒ SP
EXCHANGES									
↓ LS	MS⇒	8	9	A	B	C	D	E	F
0		A ⇔ A	B ⇔ A	CCR ⇔ A	TMP3 _L ⇔ A \$00:A ⇒ TMP3	B ⇒ A A ⇒ B	X _L ⇔ A \$00:A ⇒ X	Y _L ⇔ A \$00:A ⇒ Y	SP _L ⇔ A \$00:A ⇒ SP
1		A ⇔ B	B ⇔ B	CCR ⇔ B	TMP3 _L ⇔ B \$FF:B ⇒ TMP3	B ⇒ B \$FF ⇒ A	X _L ⇔ B \$FF:B ⇒ X	Y _L ⇔ B \$FF:B ⇒ Y	SP _L ⇔ B \$FF:B ⇒ SP
2		A ⇔ CCR	B ⇔ CCR	CCR ⇔ CCR	TMP3 _L ⇔ CCR \$FF:CCR ⇒ TMP3	B ⇒ CCR \$FF:CCR ⇒ D	X _L ⇔ CCR \$FF:CCR ⇒ X	Y _L ⇔ CCR \$FF:CCR ⇒ Y	SP _L ⇔ CCR \$FF:CCR ⇒ SP
3		\$00:A ⇒ TMP2 TMP2 _L ⇒ A	\$00:B ⇒ TMP2 TMP2 _L ⇒ B	\$00:CCR ⇒ TMP2 TMP2 _L ⇒ CCR	TMP3 ⇔ TMP2	D ⇔ TMP2	X ⇔ TMP2	Y ⇔ TMP2	SP ⇔ TMP2
4		\$00:A ⇒ D X _L ⇒ A	\$00:B ⇒ D X _L ⇒ B	\$00:CCR ⇒ D B ⇒ CCR	TMP3 ⇔ D	D ⇔ D	X ⇔ D	Y ⇔ D	SP ⇔ D
5		\$00:A ⇒ X X _L ⇒ A	\$00:B ⇒ X X _L ⇒ B	\$00:CCR ⇒ X X _L ⇒ CCR	TMP3 ⇔ X	D ⇔ X	X ⇔ X	Y ⇔ X	SP ⇔ X
6		\$00:A ⇒ Y Y _L ⇒ A	\$00:B ⇒ Y Y _L ⇒ B	\$00:CCR ⇒ Y Y _L ⇒ CCR	TMP3 ⇔ Y	D ⇔ Y	X ⇔ Y	Y ⇔ Y	SP ⇔ Y
7		\$00:A ⇒ SP SP _L ⇒ A	\$00:B ⇒ SP SP _L ⇒ B	\$00:CCR ⇒ SP SP _L ⇒ CCR	TMP3 ⇔ SP	D ⇔ SP	X ⇔ SP	Y ⇔ SP	SP ⇔ SP

TMP2 and TMP3 registers are for factory use only.

Table A-6. Loop Primitive Postbyte Encoding (Ib)

00	A	10	A	20	A	30	A	40	A	50	A	60	A	70	A	80	A	90	A	A0	A	B0	A
DBEQ	(+)	DBEQ	(-)	DBNE	(+)	DBNE	(-)	TBEQ	(+)	TBEQ	(-)	TBNE	(+)	TBNE	(-)	IBEQ	(+)	IBEQ	(-)	IBNE	(+)	IBNE	(-)
01	B	11	B	21	B	31	B	41	B	51	B	61	B	71	B	81	B	91	B	A1	B	B1	B
DBEQ	(+)	DBEQ	(-)	DBNE	(+)	DBNE	(-)	TBEQ	(+)	TBEQ	(-)	TBNE	(+)	TBNE	(-)	IBEQ	(+)	IBEQ	(-)	IBNE	(+)	IBNE	(-)
02	—	12	—	22	—	32	—	42	—	52	—	62	—	72	—	82	—	92	—	A2	—	B2	—
03	—	13	—	23	—	33	—	43	—	53	—	63	—	73	—	83	—	93	—	A3	—	B3	—
04	D	14	D	24	D	34	D	44	D	54	D	64	D	74	D	84	D	94	D	A4	D	B4	D
DBEQ	(+)	DBEQ	(-)	DBNE	(+)	DBNE	(-)	TBEQ	(+)	TBEQ	(-)	TBNE	(+)	TBNE	(-)	IBEQ	(+)	IBEQ	(-)	IBNE	(+)	IBNE	(-)
05	X	15	X	25	X	35	X	45	X	55	X	65	X	75	X	85	X	95	X	A5	X	B5	X
DBEQ	(+)	DBEQ	(-)	DBNE	(+)	DBNE	(-)	TBEQ	(+)	TBEQ	(-)	TBNE	(+)	TBNE	(-)	IBEQ	(+)	IBEQ	(-)	IBNE	(+)	IBNE	(-)
06	Y	16	Y	26	Y	36	Y	46	Y	56	Y	66	Y	76	Y	86	Y	96	Y	A6	Y	B6	Y
DBEQ	(+)	DBEQ	(-)	DBNE	(+)	DBNE	(-)	TBEQ	(+)	TBEQ	(-)	TBNE	(+)	TBNE	(-)	IBEQ	(+)	IBEQ	(-)	IBNE	(+)	IBNE	(-)
07	SP	17	SP	27	SP	37	SP	47	SP	57	SP	67	SP	77	SP	87	SP	97	SP	A7	SP	B7	SP
DBEQ	(+)	DBEQ	(-)	DBNE	(+)	DBNE	(-)	TBEQ	(+)	TBEQ	(-)	TBNE	(+)	TBNE	(-)	IBEQ	(+)	IBEQ	(-)	IBNE	(+)	IBNE	(-)

Key to Table A-6

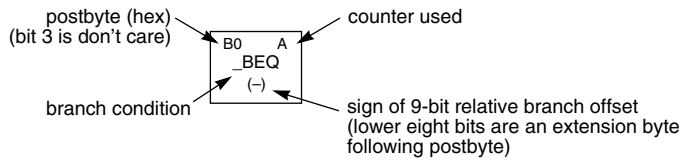


Table A-7. Branch/Complementary Branch

Branch				Complementary Branch			
Test	Mnemonic	Opcode	Boolean	Test	Mnemonic	Opcode	Comment
r>m	BGT	2E	Z + (N ⊕ V) = 0	r≤m	BLE	2F	Signed
r≥m	BGE	2C	N ⊕ V = 0	r<m	BLT	2D	Signed
r=m	BEQ	27	Z = 1	r≠m	BNE	26	Signed
r≤m	BLE	2F	Z + (N ⊕ V) = 1	r>m	BGT	2E	Signed
r<m	BLT	2D	N ⊕ V = 1	r≥m	BGE	2C	Signed
r>m	BHI	22	C + Z = 0	r≤m	BLS	23	Unsigned
r≥m	BHS/BCC	24	C = 0	r<m	BLO/BCS	25	Unsigned
r=m	BEQ	27	Z = 1	r≠m	BNE	26	Unsigned
r≤m	BLS	23	C + Z = 1	r>m	BHI	22	Unsigned
r<m	BLO/BCS	25	C = 1	r≥m	BHS/BCC	24	Unsigned
Carry	BCS	25	C = 1	No Carry	BCC	24	Simple
Negative	BMI	2B	N = 1	Plus	BPL	2A	Simple
Overflow	BVS	29	V = 1	No Overflow	BVC	28	Simple
r=0	BEQ	27	Z = 1	r≠0	BNE	26	Simple
Always	BRA	20	—	Never	BRN	21	Unconditional

For 16-bit offset long branches precede opcode with a \$18 page prebyte.

Disassembly of an HC12 Program

- It is sometimes useful to be able to convert HC12 op codes into mnemonics.
- For example, consider the hex code:

ADDR	DATA
2000	C6 05 FE 10 00 E6 01 18 06 B7 E5 04 35 EE 3F

- To determine the instructions, use Tables A.2 through A.7 of the S12CPUV2 Reference Manual.
 - If the first byte of the instruction is anything other than \$18, use Sheet 1 of Table A.2 (Page 395). From this table, determine the number of bytes of the instruction and the addressing mode. For example, \$C6 is a two-byte instruction, the mnemonic is LDAB, and it uses the IMM addressing mode. Thus, the two bytes C6 05 is the op code for the instruction LDAB #\$05.
 - The op code FE is LDX using the EXT addressing mode, and has 3 bytes total, with the last two bytes being the extended address. Thus, FE 10 00 is the instruction LDX \$1000.
 - Indexed addressing mode is fairly complicated to disassemble. You need to use Table A.3 to determine the operand. For example, the op code \$E6 indicates LDAB indexed, and may use two to four bytes (one to three bytes in addition to the op code). The postbyte 01 indicates that the operand is 0,1, which is 5-bit constant offset, which takes only one additional byte. All 5-bit constant offset, pre and post increment and decrement, and register offset instructions use one additional byte. All 9-bit constant offset instructions use two additional bytes, with the second byte holding 8 bits of the 9 bit offset. (The 9th bit is a direction bit, which is held in the first postbyte.) All 16-bit constant offset instructions use three postbytes, with the 2nd and 3rd holding the 16-bit unsigned offset.
 - If the first byte is \$18, use Sheet 2 of Table A.2 (Page 396). For example, 18 06 is a two byte instruction, the mnemonic is ABA, and it uses the INH addressing mode, so there is no operand. Thus, the two bytes 18 06 is the op code for the instruction ABA.
 - Transfer (TFR) and exchange (EXG) instructions all have the op code \$B7 with one post-byte. Use Table A.5 to determine whether it is TFR or an EXG, and to determine which registers are being used. If the most significant bit of the postbyte is 0, the instruction is a transfer instruction. B7 E5 is the code for the instruction EXG Y,X.
 - Loop instructions (*Decrement and Branch*, *Increment and Branch*, and *Test and Branch*) all have the op code \$04. To determine which instruction the op code \$04 implies, and whether the branch is positive (forward) or negative (backward), use Table A.6. For example, in the sequence 04 35 EE, the 04 indicates a loop instruction. The 35 indicates it is a DBNE X instruction (decrement register X and branch if result is not equal to zero), and the direction is backward (negative). The EE indicates a branch of -18_{10} bytes (take the 2's complement of EE).

- Use up all the bytes for one instruction, then go on to the next instruction.

ADDR DATA

 2000 C6 05 FE 10 00 E6 01 18 06 B7 E5 04 35 EE 3F

2000	C6 05	=> LDAB #\$05	two-byte LDAB, IMM addressing mode
2002	FE 10 00	=> LDX \$1000	three-byte LDX, EXT addressing mode
2005	E6 01	=> LDAB 1,X	two to four-byte LDAB, IDX addressing mode. Operand 01 => 1,X, a 5b constant offset which uses only one postbyte
2007	18 06	=> ABA	two-byte ABA, INH addressing mode
2009	B7 E5	=> EXG Y,X	two-byte exchange/transfer instruction MSB of post-byte = 1 => exchange From Table A.5, E5 => (Y <=> X)
200b	04 35 EE	=> DBNE X,(-18)	three-byte loop instruction Postbyte 35 indicates DBNE X, negative 2's complement of 0xEE is -18 decimal
200e	3F	=> SWI	one-byte SWI, INH addressing mode

Table A-2. CPU12 Opcode Map (Sheet 1 of 2)

00	f5	10	1	20	3	30	3	40	1	50	1	60	3-6	70	4	80	1	90	3	A0	3-6	B0	3	C0	1	D0	3	E0	3-6	F0	3
BGND		ANDCC		BRA		PULX		NEGA		NEGB		NEG		NEG		SUBA		SUBA		SUBA		SUBA		SUBB		SUBB		SUBB		SUBB	
IH	1	IM	2	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
01	5	11	11	21	1	31	3	41	1	51	1	61	3-6	71	4	81	1	91	3	A1	3-6	B1	3	C1	1	D1	3	E1	3-6	F1	3
MEM		EDIV		BRN		PULY		COMA		COMB		COM		COM		CMPA		CMPA		CMPA		CMPA		CMPB		CMPB		CMPB		CMPB	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
02	1	12	11	22	3/1	32	3	42	1	52	1	62	3-6	72	4	82	1	92	3	A2	3-6	B2	3	C2	1	D2	3	E2	3-6	F2	3
INY		MUL		BHI		PULA		INCA		INCB		INC		INC		SBCA		SBCA		SBCA		SBCA		SBCB		SBCB		SBCB		SBCB	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
03	1	13	3	23	3/1	33	3	43	1	53	1	63	3-6	73	4	83	2	93	3	A3	3-6	B3	3	C3	2	D3	3	E3	3-6	F3	3
DEY		EMUL		BLS		PULB		DECA		DECB		DEC		DEC		SUBD		SUBD		SUBD		SUBD		ADD		ADD		ADD		ADD	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
04	3	14	11	24	3/1	34	2	44	1	54	1	64	3-6	74	4	84	1	94	3	A4	3-6	B4	3	C4	1	D4	3	E4	3-6	F4	3
loop		ORCC		BCC		PSHX		LSRA		LSRB		LSR		LSR		ANDA		ANDA		ANDA		ANDA		ANDB		ANDB		ANDB		ANDB	
IH	1	IH	1	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
05	3-6	15	4-7	25	3/1	35	2	45	1	55	1	65	3-6	75	4	85	1	95	3	A5	3-6	B5	3	C5	1	D5	3	E5	3-6	F5	3
JMP		JSR		BCS		PSHY		ROLA		ROLB		ROL		ROL		BITA		BITA		BITA		BITA		BITB		BITB		BITB		BITB	
ID	2-4	ID	2-4	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
06	3	16	4	26	3/1	36	2	46	1	56	1	66	3-6	76	4	86	1	96	3	A6	3-6	B6	3	C6	1	D6	3	E6	3-6	F6	3
JMP		JSR		BNE		PSHA		RORA		RORB		ROR		ROR		LDAA		LDAA		LDAA		LDAA		LDAB		LDAB		LDAB		LDAB	
EX	3	EX	3	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
07	4	17	4	27	3/1	37	2	47	1	57	1	67	3-6	77	4	87	1	97	3	A7	3-6	B7	3	C7	1	D7	3	E7	3-6	F7	3
BSR		JSR		BEG		PSHB		ASRA		ASRB		ASR		ASR		CLRA		TSTA		NOP		TFR/EXG		CLRB		TSTB		TST		TST	
RL	2	DI	2	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IH	1	IH	1	IH	1	IH	1	IH	1	IH	1	ID	2-4	EX	3
08	1	18	1	28	3/1	38	3	48	1	58	1	68	3-6	78	4	88	1	98	3	A8	3-6	B8	3	C8	1	D8	3	E8	3-6	F8	3
INX		Page 2		BVC		PULC		ASLA		ASLB		ASL		ASL		EORA		EORA		EORA		EORA		EORB		EORB		EORB		EORB	
IH	1			RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
09	1	19	2	29	3/1	39	2	49	1	59	1	69	12-4	79	3	89	1	99	3	A9	3-6	B9	3	C9	1	D9	3	E9	3-6	F9	3
DEX		LEAY		BVS		PSHC		LSRD		ASLD		CLR		CLR		ADCA		ADCA		ADCA		ADCA		ADCB		ADCB		ADCB		ADCB	
IH	1	ID	2-4	RL	2	IH	1	IH	1	IH	1	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
0A	17	1A	2	2A	3/1	3A	3	4A	17	5A	2	6A	12-4	7A	3	8A	1	9A	3	AA	3-6	BA	3	CA	1	DA	3	EA	3-6	FA	3
RTC		LEAX		BPL		PULD		CALL		STAA		STAA		STAA		ORAA		ORAA		ORAA		ORAA		ORAB		ORAB		ORAB		ORAB	
IH	1	ID	2-4	RL	2	IH	1	EX	4	DI	2	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
0B	18	1B	2	2B	3/1	3B	2	4B	17-10	5B	2	6B	12-4	7B	3	8B	1	9B	3	AB	3-6	BB	3	CB	1	DB	3	EB	3-6	FB	3
RTI		LEAS		BMI		PSHD		CALL		STAB		STAB		STAB		ADDA		ADDA		ADDA		ADDA		ADDB		ADDB		ADDB		ADDB	
IH	1	ID	2-4	RL	2	IH	1	ID	2-5	DI	2	ID	2-4	EX	3	IM	2	DI	2	DI	2	DI	2	IM	2	DI	2	DI	2	DI	2
0C	4-6	1C	4	2C	3/1	3C	1+5	4C	4	5C	2	6C	12-4	7C	3	8C	2	9C	3	AC	3-6	BC	3	CC	2	DC	3	EC	3-6	FC	3
BSET		BSET		BGE		wavr		BSET		STD		STD		STD		CPD		CPD		CPD		CPD		LDD		LDD		LDD		LDD	
ID	3-5	EX	4	RL	2	SP	1	DI	3	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
0D	4-6	1D	4	2D	3/1	3D	5	4D	4	5D	2	6D	12-4	7D	3	8D	2	9D	3	AD	3-6	BD	3	CD	2	DD	3	ED	3-6	FD	3
BCLR		BCLR		BLT		RTS		BCLR		STY		STY		STY		CPY		CPY		CPY		CPY		LDY		LDY		LDY		LDY	
ID	3-5	EX	4	RL	2	IH	1	DI	3	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
0E	14-6	1E	5	2E	3/1	3E	1+7	4E	4	5E	2	6E	12-4	7E	3	8E	2	9E	3	AE	3-6	BE	3	CE	2	DE	3	EE	3-6	FE	3
BRSET		BRSET		BGT		WAI		BRSET		STX		STX		STX		CPX		CPX		CPX		CPX		LDX		LDX		LDX		LDX	
ID	4-6	EX	5	RL	2	IH	1	DI	4	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2
0F	14-6	1F	5	2F	3/1	3F	9	4F	4	5F	2	6F	12-4	7F	3	8F	2	9F	3	AF	3-6	BF	3	CF	2	DF	3	EF	3-6	FF	3
BRCLR		BRCLR		BLE		SWI		BRCLR		STS		STS		STS		CPS		CPS		CPS		CPS		LDS		LDS		LDS		LDS	
ID	4-6	EX	5	RL	2	IH	1	DI	4	DI	2	ID	2-4	EX	3	IM	3	DI	2	DI	2	DI	2	IM	3	DI	2	DI	2	DI	2

Key to Table A-2

- Opcode → 00 5 ← Number of HCS12 cycles (‡ indicates HC12 different)
- Mnemonic → BGND ←
- Address Mode → IH 1 ← Number of bytes

Table A-2. CPU12 Opcode Map (Sheet 2 of 2)

00	4	10	12	20	4	30	10	40	10	50	10	60	10	70	10	80	10	90	10	A0	10	B0	10	C0	10	D0	10	E0	10	F0	10
MOVW		IDIV		LBRA		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-ID	5	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
01	5	11	12	21	3	31	10	41	10	51	10	61	10	71	10	81	10	91	10	A1	10	B1	10	C1	10	D1	10	E1	10	F1	10
MOVW		FDIV		LBRN		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-ID	5	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
02	5	12	13	22	4/3	32	10	42	10	52	10	62	10	72	10	82	10	92	10	A2	10	B2	10	C2	10	D2	10	E2	10	F2	10
MOVW		EMACS		LBHI		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-ID	4	SP	4	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
03	5	13	3	23	4/3	33	10	43	10	53	10	63	10	73	10	83	10	93	10	A3	10	B3	10	C3	10	D3	10	E3	10	F3	10
MOVW		EMULS		LBSL		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-EX	6	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
04	6	14	12	24	4/3	34	10	44	10	54	10	64	10	74	10	84	10	94	10	A4	10	B4	10	C4	10	D4	10	E4	10	F4	10
MOVW		EDIVS		LBCC		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-EX	6	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
05	5	15	12	25	4/3	35	10	45	10	55	10	65	10	75	10	85	10	95	10	A5	10	B5	10	C5	10	D5	10	E5	10	F5	10
MOVW		IDIVS		LBCS		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-EX	5	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
06	2	16	2	26	4/3	36	10	46	10	56	10	66	10	76	10	86	10	96	10	A6	10	B6	10	C6	10	D6	10	E6	10	F6	10
ABA		SBA		LBNE		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
07	3	17	2	27	4/3	37	10	47	10	57	10	67	10	77	10	87	10	97	10	A7	10	B7	10	C7	10	D7	10	E7	10	F7	10
DAA		CBA		LBEO		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	IH	2	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
08	4	18	4-7	28	4/3	38	10	48	10	58	10	68	10	78	10	88	10	98	10	A8	10	B8	10	C8	10	D8	10	E8	10	F8	10
MOVW		MAXA		LBVC		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-ID	4	ID	3-5	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
09	5	19	4-7	29	4/3	39	10	49	10	59	10	69	10	79	10	89	10	99	10	A9	10	B9	10	C9	10	D9	10	E9	10	F9	10
MOVW		MINA		LBVS		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-ID	5	ID	3-5	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0A	5	1A	4-7	2A	4/3	3A	13n	4A	10	5A	10	6A	10	7A	10	8A	10	9A	10	AA	10	BA	10	CA	10	DA	10	EA	10	FA	10
MOVW		EMAXD		LBPL		REV		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-ID	4	ID	3-5	RL	4	SP	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0B	4	1B	4-7	2B	4/3	3B	15n/3n	4B	10	5B	10	6B	10	7B	10	8B	10	9B	10	AB	10	BB	10	CB	10	DB	10	EB	10	FB	10
MOVW		EMIND		LBMI		REVV		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IM-EX	5	ID	3-5	RL	4	SP	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0C	6	1C	4-7	2C	4/3	3C	17B	4C	10	5C	10	6C	10	7C	10	8C	10	9C	10	AC	10	BC	10	CC	10	DC	10	EC	10	FC	10
MOVW		MAXM		LBGE		WAV		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
EX-EX	6	ID	3-5	RL	4	SP	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0D	5	1D	4-7	2D	4/3	3D	18	4D	10	5D	10	6D	10	7D	10	8D	10	9D	10	AD	10	BD	10	CD	10	DD	10	ED	10	FD	10
MOVW		MINM		LBTL		TBL		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
ID-EX	5	ID	3-5	RL	4	ID	3	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0E	2	1E	4-7	2E	4/3	3E	18	4E	10	5E	10	6E	10	7E	10	8E	10	9E	10	AE	10	BE	10	CE	10	DE	10	EE	10	FE	10
TAB		EMAXM		LBGT		STOP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	ID	3-5	RL	4	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2
0F	2	1F	4-7	2F	4/3	3F	10	4F	10	5F	10	6F	10	7F	10	8F	10	9F	10	AF	10	BF	10	CF	10	DF	10	EF	10	FF	10
TBA		EMINM		LBLE		ETBL		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP		TRAP	
IH	2	ID	3-5	RL	4	ID	3	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2	IH	2

* The opcode \$04 (on sheet 1 of 2) corresponds to one of the loop primitive instructions DBEO, DBNE, IBEO, IBNE, TBEQ, or TBNE.
 † Refer to instruction summary for more information.
 ‡ Refer to instruction summary for different HC12 cycle count.
 Page 2: When the CPU encounters a page 2 opcode (\$18 on page 1 of the opcode map), it treats the next byte of object code as a page 2 instruction opcode.

Table A-3. Indexed Addressing Mode Postbyte Encoding (xb)

00	0,X 5b const	10	-16,X 5b const	20	1,+X pre-inc	30	1,X+ post-inc	40	0,Y 5b const	50	-16,Y 5b const	60	1,+Y pre-inc	70	1,Y+ post-inc	80	0,SP 5b const	90	-16,SP 5b const	A0	1,+SP pre-inc	B0	1,SP+ post-inc	C0	0,PC 5b const	D0	-16,PC 5b const	E0	n,X 9b const	F0	n,SP 9b const
01	1,X 5b const	11	-15,X 5b const	21	2,+X pre-inc	31	2,X+ post-inc	41	1,Y 5b const	51	-15,Y 5b const	61	2,+Y pre-inc	71	2,Y+ post-inc	81	1,SP 5b const	91	-15,SP 5b const	A1	2,+SP pre-inc	B1	2,SP+ post-inc	C1	1,PC 5b const	D1	-15,PC 5b const	E1	-n,X 9b const	F1	-n,SP 9b const
02	2,X 5b const	12	-14,X 5b const	22	3,+X pre-inc	32	3,X+ post-inc	42	2,Y 5b const	52	-14,Y 5b const	62	3,+Y pre-inc	72	3,Y+ post-inc	82	2,SP 5b const	92	-14,SP 5b const	A2	3,+SP pre-inc	B2	3,SP+ post-inc	C2	2,PC 5b const	D2	-14,PC 5b const	E2	n,X 16b const	F2	n,SP 16b const
03	3,X 5b const	13	-13,X 5b const	23	4,+X pre-inc	33	4,X+ post-inc	43	3,Y 5b const	53	-13,Y 5b const	63	4,+Y pre-inc	73	4,Y+ post-inc	83	3,SP 5b const	93	-13,SP 5b const	A3	4,+SP pre-inc	B3	4,SP+ post-inc	C3	3,PC 5b const	D3	-13,PC 5b const	E3	[n,X] 16b indir	F3	[n,SP] 16b indir
04	4,X 5b const	14	-12,X 5b const	24	5,+X pre-inc	34	5,X+ post-inc	44	4,Y 5b const	54	-12,Y 5b const	64	5,+Y pre-inc	74	5,Y+ post-inc	84	4,SP 5b const	94	-12,SP 5b const	A4	5,+SP pre-inc	B4	5,SP+ post-inc	C4	4,PC 5b const	D4	-12,PC 5b const	E4	A,X A offset	F4	A,SP A offset
05	5,X 5b const	15	-11,X 5b const	25	6,+X pre-inc	35	6,X+ post-inc	45	5,Y 5b const	55	-11,Y 5b const	65	6,+Y pre-inc	75	6,Y+ post-inc	85	5,SP 5b const	95	-11,SP 5b const	A5	6,+SP pre-inc	B5	6,SP+ post-inc	C5	5,PC 5b const	D5	-11,PC 5b const	E5	B,X B offset	F5	B,SP B offset
06	6,X 5b const	16	-10,X 5b const	26	7,+X pre-inc	36	7,X+ post-inc	46	6,Y 5b const	56	-10,Y 5b const	66	7,+Y pre-inc	76	7,Y+ post-inc	86	6,SP 5b const	96	-10,SP 5b const	A6	7,+SP pre-inc	B6	7,SP+ post-inc	C6	6,PC 5b const	D6	-10,PC 5b const	E6	D,X D offset	F6	D,SP D offset
07	7,X 5b const	17	-9,X 5b const	27	8,+X pre-inc	37	8,X+ post-inc	47	7,Y 5b const	57	-9,Y 5b const	67	8,+Y pre-inc	77	8,Y+ post-inc	87	7,SP 5b const	97	-9,SP 5b const	A7	8,+SP pre-inc	B7	8,SP+ post-inc	C7	7,PC 5b const	D7	-9,PC 5b const	E7	[D,X] D indirect	F7	[D,SP] D indirect
08	8,X 5b const	18	-8,X 5b const	28	8,-X pre-dec	38	8,X- post-dec	48	8,Y 5b const	58	-8,Y 5b const	68	8,-Y pre-dec	78	8,Y- post-dec	88	8,SP 5b const	98	-8,SP 5b const	A8	8,-SP pre-dec	B8	8,SP- post-dec	C8	8,PC 5b const	D8	-8,PC 5b const	E8	n,Y 9b const	F8	n,PC 9b const
09	9,X 5b const	19	-7,X 5b const	29	7,-X pre-dec	39	7,X- post-dec	49	9,Y 5b const	59	-7,Y 5b const	69	7,-Y pre-dec	79	7,Y- post-dec	89	9,SP 5b const	99	-7,SP 5b const	A9	7,-SP pre-dec	B9	7,SP- post-dec	C9	9,PC 5b const	D9	-7,PC 5b const	E9	-n,Y 9b const	F9	-n,PC 9b const
0A	10,X 5b const	1A	-6,X 5b const	2A	6,-X pre-dec	3A	6,X- post-dec	4A	10,Y 5b const	5A	-6,Y 5b const	6A	6,-Y pre-dec	7A	6,Y- post-dec	8A	10,SP 5b const	9A	-6,SP 5b const	AA	6,-SP pre-dec	BA	6,SP- post-dec	CA	10,PC 5b const	DA	-6,PC 5b const	EA	n,Y 16b const	FA	n,PC 16b const
0B	11,X 5b const	1B	-5,X 5b const	2B	5,-X pre-dec	3B	5,X- post-dec	4B	11,Y 5b const	5B	-5,Y 5b const	6B	5,-Y pre-dec	7B	5,Y- post-dec	8B	11,SP 5b const	9B	-5,SP 5b const	AB	5,-SP pre-dec	BB	5,SP- post-dec	CB	11,PC 5b const	DB	-5,PC 5b const	EB	[n,Y] 16b indir	FB	[n,PC] 16b indir
0C	12,X 5b const	1C	-4,X 5b const	2C	4,-X pre-dec	3C	4,X- post-dec	4C	12,Y 5b const	5C	-4,Y 5b const	6C	4,-Y pre-dec	7C	4,Y- post-dec	8C	12,SP 5b const	9C	-4,SP 5b const	AC	4,-SP pre-dec	BC	4,SP- post-dec	CC	12,PC 5b const	DC	-4,PC 5b const	EC	A,Y A offset	FC	A,PC A offset
0D	13,X 5b const	1D	-3,X 5b const	2D	3,-X pre-dec	3D	3,X- post-dec	4D	13,Y 5b const	5D	-3,Y 5b const	6D	3,-Y pre-dec	7D	3,Y- post-dec	8D	13,SP 5b const	9D	-3,SP 5b const	AD	3,-SP pre-dec	BD	3,SP- post-dec	CD	13,PC 5b const	DD	-3,PC 5b const	ED	B,Y B offset	FD	B,PC B offset
0E	14,X 5b const	1E	-2,X 5b const	2E	2,-X pre-dec	3E	2,X- post-dec	4E	14,Y 5b const	5E	-2,Y 5b const	6E	2,-Y pre-dec	7E	2,Y- post-dec	8E	14,SP 5b const	9E	-2,SP 5b const	AE	2,-SP pre-dec	BE	2,SP- post-dec	CE	14,PC 5b const	DE	-2,PC 5b const	EE	D,Y D offset	FE	D,PC D offset
0F	15,X 5b const	1F	-1,X 5b const	2F	1,-X pre-dec	3F	1,X- post-dec	4F	15,Y 5b const	5F	-1,Y 5b const	6F	1,-Y pre-dec	7F	1,Y- post-dec	8F	15,SP 5b const	9F	-1,SP 5b const	AF	1,-SP pre-dec	BF	1,SP- post-dec	CF	15,PC 5b const	DF	-1,PC 5b const	EF	[D,Y] D indirect	FF	[D,PC] D indirect

Key to Table A-3

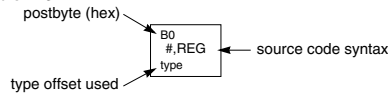


Table A-5. Transfer and Exchange Postbyte Encoding

TRANSFERS									
↓ LS	MS⇒	0	1	2	3	4	5	6	7
0		A ⇒ A	B ⇒ A	CCR ⇒ A	TMP3 _L ⇒ A	B ⇒ A	X _L ⇒ A	Y _L ⇒ A	SP _L ⇒ A
1		A ⇒ B	B ⇒ B	CCR ⇒ B	TMP3 _L ⇒ B	B ⇒ B	X _L ⇒ B	Y _L ⇒ B	SP _L ⇒ B
2		A ⇒ CCR	B ⇒ CCR	CCR ⇒ CCR	TMP3 _L ⇒ CCR	B ⇒ CCR	X _L ⇒ CCR	Y _L ⇒ CCR	SP _L ⇒ CCR
3		sex:A ⇒ TMP2	sex:B ⇒ TMP2	sex:CCR ⇒ TMP2	TMP3 ⇒ TMP2	D ⇒ TMP2	X ⇒ TMP2	Y ⇒ TMP2	SP ⇒ TMP2
4		sex:A ⇒ D SEX A,D	sex:B ⇒ D SEX B,D	sex:CCR ⇒ D SEX CCR,D	TMP3 ⇒ D	D ⇒ D	X ⇒ D	Y ⇒ D	SP ⇒ D
5		sex:A ⇒ X SEX A,X	sex:B ⇒ X SEX B,X	sex:CCR ⇒ X SEX CCR,X	TMP3 ⇒ X	D ⇒ X	X ⇒ X	Y ⇒ X	SP ⇒ X
6		sex:A ⇒ Y SEX A,Y	sex:B ⇒ Y SEX B,Y	sex:CCR ⇒ Y SEX CCR,Y	TMP3 ⇒ Y	D ⇒ Y	X ⇒ Y	Y ⇒ Y	SP ⇒ Y
7		sex:A ⇒ SP SEX A,SP	sex:B ⇒ SP SEX B,SP	sex:CCR ⇒ SP SEX CCR,SP	TMP3 ⇒ SP	D ⇒ SP	X ⇒ SP	Y ⇒ SP	SP ⇒ SP
EXCHANGES									
↓ LS	MS⇒	8	9	A	B	C	D	E	F
0		A ⇔ A	B ⇔ A	CCR ⇔ A	TMP3 _L ⇔ A \$00:A ⇒ TMP3	B ⇒ A A ⇒ B	X _L ⇔ A \$00:A ⇒ X	Y _L ⇔ A \$00:A ⇒ Y	SP _L ⇔ A \$00:A ⇒ SP
1		A ⇔ B	B ⇔ B	CCR ⇔ B	TMP3 _L ⇔ B \$FF:B ⇒ TMP3	B ⇒ B \$FF ⇒ A	X _L ⇔ B \$FF:B ⇒ X	Y _L ⇔ B \$FF:B ⇒ Y	SP _L ⇔ B \$FF:B ⇒ SP
2		A ⇔ CCR	B ⇔ CCR	CCR ⇔ CCR	TMP3 _L ⇔ CCR \$FF:CCR ⇒ TMP3	B ⇒ CCR \$FF:CCR ⇒ D	X _L ⇔ CCR \$FF:CCR ⇒ X	Y _L ⇔ CCR \$FF:CCR ⇒ Y	SP _L ⇔ CCR \$FF:CCR ⇒ SP
3		\$00:A ⇒ TMP2 TMP2 _L ⇒ A	\$00:B ⇒ TMP2 TMP2 _L ⇒ B	\$00:CCR ⇒ TMP2 TMP2 _L ⇒ CCR	TMP3 ⇔ TMP2	D ⇔ TMP2	X ⇔ TMP2	Y ⇔ TMP2	SP ⇔ TMP2
4		\$00:A ⇒ D X _L ⇒ A	\$00:B ⇒ D X _L ⇒ B	\$00:CCR ⇒ D B ⇒ CCR	TMP3 ⇔ D	D ⇔ D	X ⇔ D	Y ⇔ D	SP ⇔ D
5		\$00:A ⇒ X Y _L ⇒ A	\$00:B ⇒ X Y _L ⇒ B	\$00:CCR ⇒ X X _L ⇒ CCR	TMP3 ⇔ X	D ⇔ X	X ⇔ X	Y ⇔ X	SP ⇔ X
6		\$00:A ⇒ Y SP _L ⇒ A	\$00:B ⇒ Y SP _L ⇒ B	\$00:CCR ⇒ Y SP _L ⇒ CCR	TMP3 ⇔ Y	D ⇔ Y	X ⇔ Y	Y ⇔ Y	SP ⇔ Y
7		\$00:A ⇒ SP SP _L ⇒ A	\$00:B ⇒ SP SP _L ⇒ B	\$00:CCR ⇒ SP SP _L ⇒ CCR	TMP3 ⇔ SP	D ⇔ SP	X ⇔ SP	Y ⇔ SP	SP ⇔ SP

TMP2 and TMP3 registers are for factory use only.

Table A-6. Loop Primitive Postbyte Encoding (Ib)

00 DBEQ (+) A	10 DBEQ (-) A	20 DBNE (+) A	30 DBNE (-) A	40 TBEQ (+) A	50 TBEQ (-) A	60 TBNE (+) A	70 TBNE (-) A	80 IBEQ (+) A	90 IBEQ (-) A	A0 IBNE (+) A	B0 IBNE (-) A
01 DBEQ (+) B	11 DBEQ (-) B	21 DBNE (+) B	31 DBNE (-) B	41 TBEQ (+) B	51 TBEQ (-) B	61 TBNE (+) B	71 TBNE (-) B	81 IBEQ (+) B	91 IBEQ (-) B	A1 IBNE (+) B	B1 IBNE (-) B
02 —	12 —	22 —	32 —	42 —	52 —	62 —	72 —	82 —	92 —	A2 —	B2 —
03 —	13 —	23 —	33 —	43 —	53 —	63 —	73 —	83 —	93 —	A3 —	B3 —
04 DBEQ (+) D	14 DBEQ (-) D	24 DBNE (+) D	34 DBNE (-) D	44 TBEQ (+) D	54 TBEQ (-) D	64 TBNE (+) D	74 TBNE (-) D	84 IBEQ (+) D	94 IBEQ (-) D	A4 IBNE (+) D	B4 IBNE (-) D
05 DBEQ (+) X	15 DBEQ (-) X	25 DBNE (+) X	35 DBNE (-) X	45 TBEQ (+) X	55 TBEQ (-) X	65 TBNE (+) X	75 TBNE (-) X	85 IBEQ (+) X	95 IBEQ (-) X	A5 IBNE (+) X	B5 IBNE (-) X
06 DBEQ (+) Y	16 DBEQ (-) Y	26 DBNE (+) Y	36 DBNE (-) Y	46 TBEQ (+) Y	56 TBEQ (-) Y	66 TBNE (+) Y	76 TBNE (-) Y	86 IBEQ (+) Y	96 IBEQ (-) Y	A6 IBNE (+) Y	B6 IBNE (-) Y
07 DBEQ (+) SP	17 DBEQ (-) SP	27 DBNE (+) SP	37 DBNE (-) SP	47 TBEQ (+) SP	57 TBEQ (-) SP	67 TBNE (+) SP	77 TBNE (-) SP	87 IBEQ (+) SP	97 IBEQ (-) SP	A7 IBNE (+) SP	B7 IBNE (-) SP

Key to Table A-6

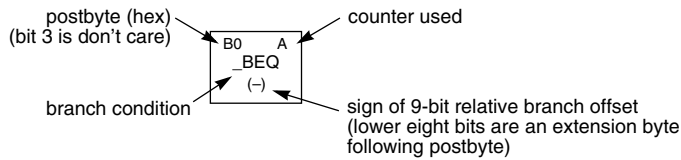


Table A-7. Branch/Complementary Branch

Branch				Complementary Branch			
Test	Mnemonic	Opcode	Boolean	Test	Mnemonic	Opcode	Comment
r>m	BGT	2E	Z + (N ⊕ V) = 0	r≤m	BLE	2F	Signed
r≥m	BGE	2C	N ⊕ V = 0	r<m	BLT	2D	Signed
r=m	BEQ	27	Z = 1	r≠m	BNE	26	Signed
r≤m	BLE	2F	Z + (N ⊕ V) = 1	r>m	BGT	2E	Signed
r<m	BLT	2D	N ⊕ V = 1	r≥m	BGE	2C	Signed
r>m	BHI	22	C + Z = 0	r≤m	BLS	23	Unsigned
r≥m	BHS/BCC	24	C = 0	r<m	BLO/BCS	25	Unsigned
r=m	BEQ	27	Z = 1	r≠m	BNE	26	Unsigned
r≤m	BLS	23	C + Z = 1	r>m	BHI	22	Unsigned
r<m	BLO/BCS	25	C = 1	r≥m	BHS/BCC	24	Unsigned
Carry	BCS	25	C = 1	No Carry	BCC	24	Simple
Negative	BMI	2B	N = 1	Plus	BPL	2A	Simple
Overflow	BVS	29	V = 1	No Overflow	BVC	28	Simple
r=0	BEQ	27	Z = 1	r≠0	BNE	26	Simple
Always	BRA	20	—	Never	BRN	21	Unconditional

For 16-bit offset long branches precede opcode with a \$18 page prebyte.

Binary, Hex and Decimal Numbers (4-bit representation)

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

What does a number represent?

Binary numbers are a code, and represent what the programme intends for the code.

0x72 Some possible codes:

`'r'` (ASCII)

`INC` (HC12 instruction)

`2.26V` (Input from A/D converter)

`11410` (Unsigned 8-bit number)

`+11410` (Signed 8-bit number)

`Set temperature in room to 69 F`

`Set cruise control speed to 120 mph`

Binary to Unsigned Decimal:**Convert Binary to Unsigned Decimal**1111011₂

$$1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$1 \times 64 + 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$$


123₁₀Hex to Unsigned Decimal**Convert Hex to Unsigned Decimal**82D6₁₆

$$8 \times 16^3 + 2 \times 16^2 + 13 \times 16^1 + 6 \times 16^0$$

$$8 \times 4096 + 2 \times 256 + 13 \times 16 + 6 \times 1$$

33494₁₀

Unsigned Decimal to Hex**Convert Unsigned Decimal to Hex**

Division	Q	R	
		Decimal	Hex
721/16	45	1	1 
45/16	2	13	D
2/16	0	2	2

$$721_{10} = 2D1_{16}$$

Signed Number Representation in 2's Complement Form:

If most significant bit is 0 (most significant hex digit 0-7), number is positive.
Get decimal equivalent by converting number to decimal, and using + sign.

Example for 8-bit number:

$$\begin{aligned} 3A_{16} &\rightarrow + (3 \times 16^1 + 10 \times 16^0)_{10} \\ &\quad + (3 \times 16 + 10 \times 1)_{10} \\ &\quad + 58_{10} \end{aligned}$$

If most significant bit is 1 (most significant hex digit 8-F), number is negative.
Get decimal equivalent by taking 2's complement of number, converting to decimal, and using - sign.

Example for 8-bit number:

$$\begin{aligned} A3_{16} &\rightarrow - (5D)_{16} \\ &\quad - (5 \times 16^1 + 13 \times 16^0)_{10} \\ &\quad - (5 \times 16 + 13 \times 1)_{10} \\ &\quad - 93_{10} \end{aligned}$$

One's Complement Table Makes It Simple To Find 2's Complements

One's Complement Table

0	F
1	E
2	D
3	C
4	B
5	A
6	9
7	8

To take two's complement, add one to one's complement

Take two's complement of D0C3 :

$$2F3C + 1 = 2F3D$$

Addition and Subtraction of Hexadecimal Numbers.

Setting the C (Carry), V (Overflow), N (Negative) and Z (Zero) bits

How the C, V, N and Z bits of the CCR are changed

Condition Code Register Bits N, Z, V, C

N bit is set if result of operation is negative (MSB = 1)

Z bit is set if result of operation is zero (All bits = 0)

V bit is set if operation produced an overflow

C bit is set if operation produced a carry (borrow on subtraction)

Note: Not all instructions change these bits of the CCR

Addition of Hexadecimal Numbers**ADDITION:**

C bit set when result does not fit in word

V bit set when $P + P = N$
 $N + N = P$

N bit set when MSB of result is 1

Z bit set when result is 0

<u>7A</u>	<u>2A</u>	<u>AC</u>	<u>AC</u>
<u>+52</u>	<u>+52</u>	<u>+8A</u>	<u>+72</u>
CC	7C	36	1E
C: 0	C: 0	C: 1	C: 1
V: 1	V: 0	V: 1	V: 0
N: 1	N: 0	N: 0	N: 1
Z: 0	Z: 0	Z: 0	Z: 0

Subtraction of Hexadecimal Numbers**SUBTRACTION:**

**C bit set on borrow (when the magnitude of the subtrahend
is greater than the minuend)**

**V bit set when $N - P = P$
 $P - N = N$**

N bit set when MSB is 1

Z bit set when result is 0

$\begin{array}{r} 7A \\ -5C \\ \hline 1E \end{array}$	$\begin{array}{r} 8A \\ -5C \\ \hline 2E \end{array}$	$\begin{array}{r} 5C \\ -8A \\ \hline D2 \end{array}$	$\begin{array}{r} 2C \\ -72 \\ \hline BA \end{array}$
C: 0	C: 0	C: 1	C: 1
V: 0	V: 1	V: 1	V: 0
N: 0	N: 0	N: 1	N: 1
Z: 0	Z: 0	Z: 0	Z: 0